

國立清華大學

智慧化企業整合

停車位預約系統流程改善

指導教授:邱銘傳 教授

組別:第 3 組

組員: 112034553 黃柏欽

112034560 劉倫華

112034565 黃琬芹

112034566 洪庭緣

中華民國一一三年六月十四

目錄

一、研究動機與現況分析	7
1.1 研究動機.....	7
1.2 現況分析-5W1H.....	7
二、研究方法	9
2.1 LSTM.....	9
2.2 BiLSTM	9
2.3 TSD-GRU	10
2.3.1 時間序列分解 (TSD)	10
2.3.2 門控迴圈單元 (GRU)	10
三、模型建立	12
3.1 資料集介紹.....	12
3.2 模型建立.....	14
3.2.1 LSTM.....	14
3.2.2 BiLSTM	15
3.2.3 TSD-GRU	16
3.3 結果比較.....	18
四、網頁設計與後臺管理系統	19
4.1 Web 架構圖	19

4.2 ER-Model.....	20
4.3 功能介紹.....	21
4.3.1 網路預約及查詢.....	21
4.3.2 會員專區.....	22
4.3.3 後台管理.....	24
五、結論與未來展望	26
5.1 改善成果.....	26
5.2 未來展望.....	26

圖目錄

圖 1、Calgary 停車場分布圖.....	12
圖 2、Calgary 交通流量圖	12
圖 3、影響 occupancy 因素圖	13
圖 4、Calgary 停車數據集.....	14
圖 5、LSTM 定義超參數.....	14
圖 6、LSTM 輸入數據.....	15
圖 7、BiLSTM 定義超參數.....	15
圖 8、BiLSTM 輸入數據.....	16
圖 9、TSD-preparation	16
圖 10、Customize GRU.....	17
圖 11、TSD-GRU model creation.....	18
圖 12、網頁架構圖	20
圖 13、ER-Model	21
圖 14、即時停車場資訊介面	22
圖 15、車位預約介面	22
圖 16、會員預約介面	23
圖 17、會員預約資料填寫	24
圖 18、預約成功確認信	24

圖 19、停車場資料列表	25
--------------------	----

表目錄

表 1、模型結果.....	18
---------------	----

一、研究動機與現況分析

1.1 研究動機

在清華大學停車場，以往車輛主要依賴人工尋找車位，這種方式效率低且容易導致停車時間延長，在缺乏有效管理系統的情況下，這不僅耗費了大量的時間，也增加了停車場的混亂程度。尤其在高峰時段，車輛難以找到合適位置停放，導致塞車及浪費時間，並影響了校園內的交通流暢度和生活品質。

為了解決這一問題，本團隊創建一個網站可以查詢車位及線上繳費系統，停車用戶在停車時可以更加便捷快速、縮短流程時間。使用者可以通過網頁或應用程式查詢停車場的實時車位情況，並且在需要時即可預訂停車位，從而節省了尋找車位的時間。同時，線上繳費功能也使得停車過程更加快捷方便，避免了排隊等待的情況。這一改善措施不僅提升了停車場的運營效率，也提高了用戶的停車體驗，為清華大學停車管理帶來了更好的服務品質。

1.2 現況分析-5W1H

WHO 清大停車場用戶，如學生、老師、附近居民等

WHAT 該項目旨在預測停車場的佔用率，以便實施動態停車定價，減少尖峰時段的擁堵，並鼓勵使用附近的替代停車設施。

WHEN 平日上班上課期間

WHERE 清大校園內 4 處停車場

WHY 人工尋找車位耗時且無法有效利用停車位，隨著都市化進程的推進和私人汽車擁有量的增加，停車位難找的問題日益嚴重。實施動態停車定價可以緩解停車場尖峰時段的擁堵，優化停車資源的利用，並減少尋找停車位所造成的交通擁堵。

HOW 價值溪流圖 VSM、限制理論 TOC，並採用了 TSD-GRU 模型進行資料訓練，透過時間序列分解方法 (TSD) 和門控循環單元 (GRU) 來預測停車佔用率，並與其他模型 (如 GRU、LSTM、BiLSTM) 進行了比較，證明了 TSD-GRU 模型在預測精度上的精確性。

二、研究方法

本文將採用 3 種模型進行測試，包含 LSTM、BiLSTM 及 TSD-GRU，下列小節將逐一針對各項模型進行說明比對。

2.1 LSTM

LSTM (Long Short-Term Memory) 是一種特殊的循環神經網絡 (Recurrent Neural Network, RNN)，其主要目的為解決 RNN 會有梯度消失等問題，可有效克服在較常序列訓練過程中無法充分發揮表現的疑慮。相比於傳統的 RNN，LSTM 通過引入門控機制，可以更好地捕捉和保持序列中的長期相關性。LSTM 模型的結構包含了一個或多個 LSTM 層，每層都有一個細胞狀態 (cell state) 和一個隱藏狀態 (hidden state)。細胞狀態負責記住長期的信息，而隱藏狀態則根據當前輸入和前一個隱藏狀態進行計算，提供模型的預測結果。在每個時間，模型根據當前輸入和目標輸出計算預測值，然後通過比較預測值和真實值的誤差來更新模型的權重，以使預測結果更接近真實結果。

2.2 BiLSTM

雙向長短時記憶網路 (Bidirectional Long Short-Term Memory, BiLSTM) 類似於 LSTM，在傳統 LSTM 的基礎上引入了雙向傳播機制，同時捕獲過去和未來的上下文。BiLSTM 由兩個獨立的 LSTM 層組成：前向 LSTM 層 (Forward LSTM)：前向 LSTM 層接收輸入序列並按照時間順序 (從前到後) 處理，每個時間產生一個隱藏狀態；後向 LSTM 層 (Backward LSTM)：後向 LSTM 層接收

輸入序列並按照時間逆序 (從後到前) 處理，每個時間也產生一個隱藏狀態。將前向和後向的隱藏狀態進行連接，合併後的隱藏狀態傳遞到輸出層，進行預測或分類任務。BiLSTM 通過結合前向和後向的上下文信息，顯著增強了對序列數據的理解和處理能力，是處理許多複雜序列任務的強大工具。

2.3 TSD-GRU

TSD-GRU (時間序列分解 - 閘控循環單元) 模型是一種專為時間序列數據預測而設計的模型，在閘控迴圈單元 (GRU) 模型中應用時間序列分解 (TSD) 數據，將輸入數據分解為趨勢、季節性和殘差分量，然後再將其輸入到 GRU 模型中。此結構的目的是有效地捕獲和解釋時間序列數據中典型的重複模式和不規則波動。模型分為兩個主要部分：

2.3.1 時間序列分解 (TSD)

該元件對於塑造有關停車佔用率的時間序列數據至關重要。TSD 的作用是将數據分解為三個基本元素：趨勢、週期和效果。這些元素有助於了解數據隨時間推移的基本模式和變化。

2.3.2 閘控迴圈單元 (GRU)

GRU 部分負責構建網路框架，該框架處理由趨勢、週期和效應特徵定義的輸入。GRU 的架構與存在梯度消失問題的 RNN 不同，GRU 通過其更新和重置閘來緩解這個問題，從而在更長的序列中更好地保留資訊；與 LSTM 相比，GRU

需要的參數更少，不僅訓練速度更快，而且在計算資源方面也更有效率。

TSD 元件將數據分解為趨勢、週期和效應特徵，而 GRU 模型則根據這些特徵預測未來值，應用 TSD 然後將此數據饋送到 GRU 模型中。它包括一個訓練迴圈，其中模型的參數在每個週期中隨機初始化，並在多個週期內計算和最小化損失。這兩種技術的結合使 TSD-GRU 模型能夠通過先理解數據的基礎結構，然後使用強大的神經網絡進行準確的預測，在多次反覆運算評估模型在預測時間序列數據方面的總體準確性和有效性。

三、模型建立

本文將會利用上述介紹的三種模型(LSTM、BiLSTM 及 TSD-GRU)來做此次期末報告的模型使用。以下會分成資料集介紹、模型建立及結果比較來說明。

3.1 資料集介紹

「Calgary's parking」是關於加拿大 Calgary 城市的停車數據資料集，其 Calgary 城市的停車場分布圖如下圖 1，裡面包含了每五分鐘更新一次的停車空位數據、天氣狀況、附近的交通流量(如下圖 2)等。這些資料可以進行停車場空位的預測。我們參考了 2018 年 6 月 1 日到 6 月 30 日中 30 個停車場的資料，並將停車場分為 Commercial、Tourist、Residential、Hospital、Office 及 Sport 六大種類進行預測。



圖 1、Calgary 停車場分布圖



圖 2、Calgary 交通流量圖

由圖 3 可發現 occupancy 從早上 6 點開始顯著增加，直到中午達到高峰，在達到峰值後，佔用率從下午 3 點左右開始逐漸下降，並持續到晚上。週一到週五 (1-5 天) 遵循相似的模式，早上 occupancy 急劇增加，中午達到高峰，然

後在下午逐漸下降，這顯示出停車模式受工作日的影響。在週日（0）和週六（6）顯示出稍微不同的模式：週日是所有日子裡面峰值最低的一天，佔用率的上升和下降都較慢。週六（6）在 10 點到 15 點之間有一個中等的峰值，但不如工作日那麼明顯。依照此原則重複進行分析可得到 INDICATOR 的公式。公式如下所示。

$INDICATOR = \text{基礎值} + (\text{時間權重} \times \text{時間}) + (\text{日期權重} \times \text{一周中的日子}) + (\text{天氣權重} \times \text{天氣狀況}) + \text{地點調整} + \text{事件調整}$

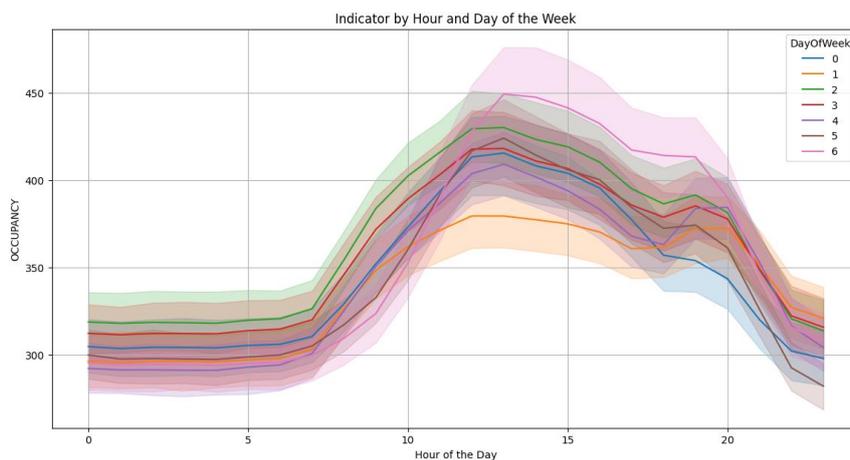


圖 3、影響 occupancy 因素圖

30 個停車場的訓練數據集包含以下內容，如圖 4 所示：

1. TIME：2018 年 6 月 1 日到 6 月 30 日，每個 5 分鐘更新。
2. OCCUPANCY: 停車場的汽車數量。
3. RATE: 停車位佔用率(停車位與停車位總數的比率)。
4. INDICATOR:根據平日假日、天氣狀況、交通流量等特徵計算出的值。
5. FS: 利用 Fourier Transform 將時間域中的數據轉換到頻率域中。可以分析數

據中的頻率成分和周期性。

	TIME	OCCUPANCY	RATE	INDICATOR	FS
0	2018/6/1 0:00	200	0.200	0.279856	0.255971
1	2018/6/1 0:05	200	0.200	0.277484	0.255497
2	2018/6/1 0:10	200	0.200	0.276493	0.255299
3	2018/6/1 0:15	202	0.202	0.264126	0.255354
4	2018/6/1 0:20	203	0.203	0.259295	0.255637

圖 4、Calgary 停車數據集

3.2 模型建立

3.2.1 LSTM

定義超參數，如圖 5 所示。

```
# Define hyperparameters for the LSTM model
LOOK_BACK = 3
INPUT_FEATURES_NUM = LOOK_BACK
HIDDEN_SIZE = 8
OUTPUT_FEATURES_NUM = 1
NUM_LAYERS = 1
max_epochs = 4000
LEARNING_RATE = 0.02
circle = 10
```

圖 5、LSTM 定義超參數

設置了 LSTM 層和全連接層，LSTM 層接受輸入特徵大小、隱藏單元數量、輸出大小和 LSTM 層數。將輸入數據經過 LSTM 層和全連接層，最終輸出預測結果，如圖 6 所示。

```

# Define the LSTM Neural Network
class LSTMModel(nn.Module) :

    def __init__(self, input_size, hidden_size, output_size, num_layers):
        super().__init__()
        self.lstm = nn.LSTM(input_size, hidden_size, num_layers) # utilize the GRU model in torch.nn
        self.forwardCalculation = nn.Linear(hidden_size, output_size)

    def forward(self, _x):
        x, _ = self.lstm(_x) # _x is input, size (seq_len, batch, input_size)
        s, b, h = x.shape # x is output, size (seq_len, batch, hidden_size)
        x = x.view(s * b, h)
        x = self.forwardCalculation(x)
        x = x.view(s, b, -1)
        return x

# Initialize the LSTM model
gru_model = LSTMModel(input_size=INPUT_FEATURES_NUM, hidden_size=HIDDEN_SIZE, output_size=OUTPUT_FEATURES_NUM, num_layers=NUM_LAYERS)
loss_function = nn.MSELoss()
optimizer = torch.optim.Adam(gru_model.parameters(), lr=LEARNING_RATE)

```

圖 6、LSTM 輸入數據

3.2.2 BiLSTM

定義超參數，如圖 7 所示。

```

# Define hyperparameters for the model
LOOK_BACK = 9
INPUT_FEATURES_NUM = LOOK_BACK
HIDDEN_SIZE = 8
OUTPUT_FEATURES_NUM = 1
NUM_LAYERS = 1
max_epochs = 500
LEARNING_RATE = 0.02
circle = 5

```

圖 7、BiLSTM 定義超參數

設置了雙向 LSTM 層和全連接層。雙向 LSTM 層接受輸入特徵大小、隱藏單元數量、輸出大小和 LSTM 層數，如圖 8 所示。

```

# Define a Bidirectional LSTM (BiLSTM) model
class BiLSTMModel(nn.Module) :
    def __init__(self, input_size, hidden_size, output_size, num_layers):
        super().__init__()
        self.lstm = nn.LSTM(input_size, hidden_size, num_layers, bidirectional=True) # utilize the GRU model in torch.nn
        self.forwardCalculation = nn.Linear(hidden_size*2, output_size)

    def forward(self, _x):
        x, _ = self.lstm(_x) # _x is input, size (seq_len, batch, input_size)
        s, b, h = x.shape # x is output, size (seq_len, batch, hidden_size)
        x = x.view(s * b, h)
        x = self.forwardCalculation(x)
        x = x.view(s, b, -1)
        return x

# Initialize the BiLSTM model
gru_model = BiLSTMModel(input_size=INPUT_FEATURES_NUM, hidden_size=HIDDEN_SIZE, output_size=OUTPUT_FEATURES_NUM, num_layers=NUM_LAYERS)
for name, param in gru_model.named_parameters():
    if param.requires_grad:
        print(param.shape)
loss_function = nn.MSELoss()
optimizer = torch.optim.Adam(gru_model.parameters(), lr=LEARNING_RATE)

```

圖 8、BiLSTM 輸入數據

3.2.3 TSD-GRU

- TSD-preparation

定義了 `import_data`，建立 `training sets` 和 `test sets`，趨勢特徵、循環特徵以及時間序列的數據集，這些函數主要用於數據的準備工作，以便後續的模型訓練和評估，如圖 9 所示。

```
# trend feature
def create_interval_dataset(dataset, look_back):
    dataX, dataY = [], []
    for i in range(len(dataset) - look_back):
        dataX.append(dataset[i])
        dataY.append(dataset[i+look_back])
    return np.asarray(dataX), np.asarray(dataY)

# cycle feature
def create_cycle_dataset(dataset, look_back):
    dataX, dataY = [], []
    for i in range(len(dataset) - look_back * 2):
        dataX.append(dataset[i:i + look_back])
        dataY.append(dataset[i + look_back:i + look_back + look_back])
    return np.asarray(dataX), np.asarray(dataY)

# RNN parameters
def create_RNNs_dataset(dataset, look_back):
    dataX, dataY = [], []
    for i in range(len(dataset) - look_back * 2):
        dataX.append(dataset[i:i + look_back])
        dataY.append(dataset[i + look_back + look_back])
    return np.asarray(dataX), np.asarray(dataY)

# import time-series-decomposition dataset
def import_TSD_data():
    file_names = [
        'Commercial1.csv', 'Commercial2.csv', 'Commercial3.csv', 'Commercial4.csv', 'Commercial5.csv',
        'Commercial6.csv', 'Commercial7.csv', 'Commercial8.csv', 'Commercial9.csv', 'Commercial10.csv',
```

圖 9、TSD-preparation

- Customize GRU

建立和訓練 GRU 模型基本框架和參數設置，用於處理時間序列數據。序列的每個元素都包含了一組特徵，首先通過 GRU 層處理了序列的特徵部分，生成了隱藏狀態。然後，這個隱藏狀態與效應特徵部分進行拼接，形成了新的特徵矩陣，並通過線性層進行處理，最終生成預測結果，如圖 10 所示。

```

# input data
Trend, Cycle, Effect, train_size, test_size = import_TSD_data()

# Hyperparameters
LOOK_BACK = 6
INPUT_FEATURES_NUM = 2*LOOK_BACK + 1
HIDDEN_SIZE = 8
OUTPUT_FEATURES_NUM = 1
NUM_LAYERS = 1
max_epochs = 4000
LEARNING_RATE = 0.02
circle = 5

# GRU Neural Networks
class GRU_model(nn.Module):
    def __init__(self, input_size, hidden_size, output_size, num_layers):
        super().__init__()
        self.gru = nn.GRU((input_size - 1), hidden_size, num_layers)
        self.forwardProcessing = nn.Linear((hidden_size + 1), output_size)

    # override initial forward
    def forward(self, x_input):
        # x_input is the input matrix
        # size (seq length, batch, input_size)
        x = x_input[:, :, 0:(INPUT_FEATURES_NUM-1)]
        e = x_input[:, :, (INPUT_FEATURES_NUM-1)].reshape(1, x_input.shape[1], 1)
        x, _ = self.gru(x)
        x = torch.cat([x, e], dim=2)
        s, b, h = x.shape
        x = x.view(s * b, h)
        x = self.forwardProcessing(x)
        x = x.view(s, b, -1)
        return x

```

圖 10、Customize GRU

- TSD-GRU model creation

它將數據集劃分為訓練集和測試集，然後對訓練數據進行 TSD 處理，提取出趨勢、周期和效應特徵，用於 GRU 模型的訓練。在模型的訓練過程中，它通過多個循環進行訓練，每個循環中都隨機初始化模型的參數。在每個循環中，模型通過訓練數據進行前向傳播和反向傳播，計算損失並更新模型參數，以最小化損失函數。這個過程重複進行多個時期，直到模型收斂或達到指定的最大時期數，如圖 11 所示。

```

from tqdm import tqdm
# TSD-GRU formulation
def TSD_GRU(gru_model, trend, cycle, effect):

    # define training and testing sets after applying Time-Series-Decomposition
    train_trend = trend[:train_size]
    test_trend = trend[train_size:]
    train_cycle = cycle[:train_size]
    test_cycle = cycle[train_size:]
    effect = effect.reshape(-1, 1)
    train_effect = effect[LOOK_BACK+LOOK_BACK:train_size]
    test_effect = effect[LOOK_BACK+LOOK_BACK+train_size:]

    # create training set
    train_trend, train_y = create_RNNs_dataset(train_trend, look_back=LOOK_BACK)
    _, train_cycle = create_cycle_dataset(train_cycle, look_back=LOOK_BACK)

    train_x = np.concatenate((train_trend, train_cycle, train_cycle), axis = 1)
    train_x = torch.tensor(train_x, dtype=torch.float32)
    train_y = torch.tensor(train_y, dtype=torch.float32)

    batch_size = train_x.shape[0]

    input_feature = train_x.shape[1]
    train_x_tensor = train_x.reshape(1, batch_size, input_feature)
    train_x_tensor = torch.tensor(train_x_tensor, dtype=torch.float32)
    train_y_tensor = train_y.reshape(1, batch_size, OUTPUT_FEATURES_NUM)
    train_y_tensor = torch.tensor(train_y_tensor, dtype=torch.float32)

    # create testing sets
    test_trend, test_y = create_RNNs_dataset(test_trend, look_back=LOOK_BACK)

```

圖 11、TSD-GRU model creation

3.3 結果比較

下表 1 為三種模型之預測結果，發現 TSD-GRU 模型結果較優，因此之後將使用 TSD-GRU 模型進行停車位預測。

表 1、模型結果

Model	MAPE(%)	RMSE($\times 10^{-2}$)	RAE(%)	R ² (%)
LSTM	3.54	1.69	15.6	93.3
BiLSTM	5.00	3.74	21.4	94.1
TSD-GRU	2.33	1.66	12.35	96.4

四、網頁設計與後臺管理系統

4.1 Web 架構圖

本網頁設計架構如圖 12 所示，主要分為網站基本功能及會員功能，網站基本功能可提供首頁(服務介紹、成員介紹、Chatbot、外部連結)、停車場介紹、停車位預約&查詢、會員登入及聯絡我們之相關功能。會員功能包括加入會員、密碼找回、修改刪除會員資料、車位預約、修改刪除預約、線上繳費。

另外，透過管理員登入讓管理員帳號能進入後台系統，有預約資料管理、會員資料管理、管理員資料管理及停車場資料管理四大功能區塊，皆可以查看現有的資訊，並直接對資料庫進行修改刪除(管理員管理額外提供新增功能)。

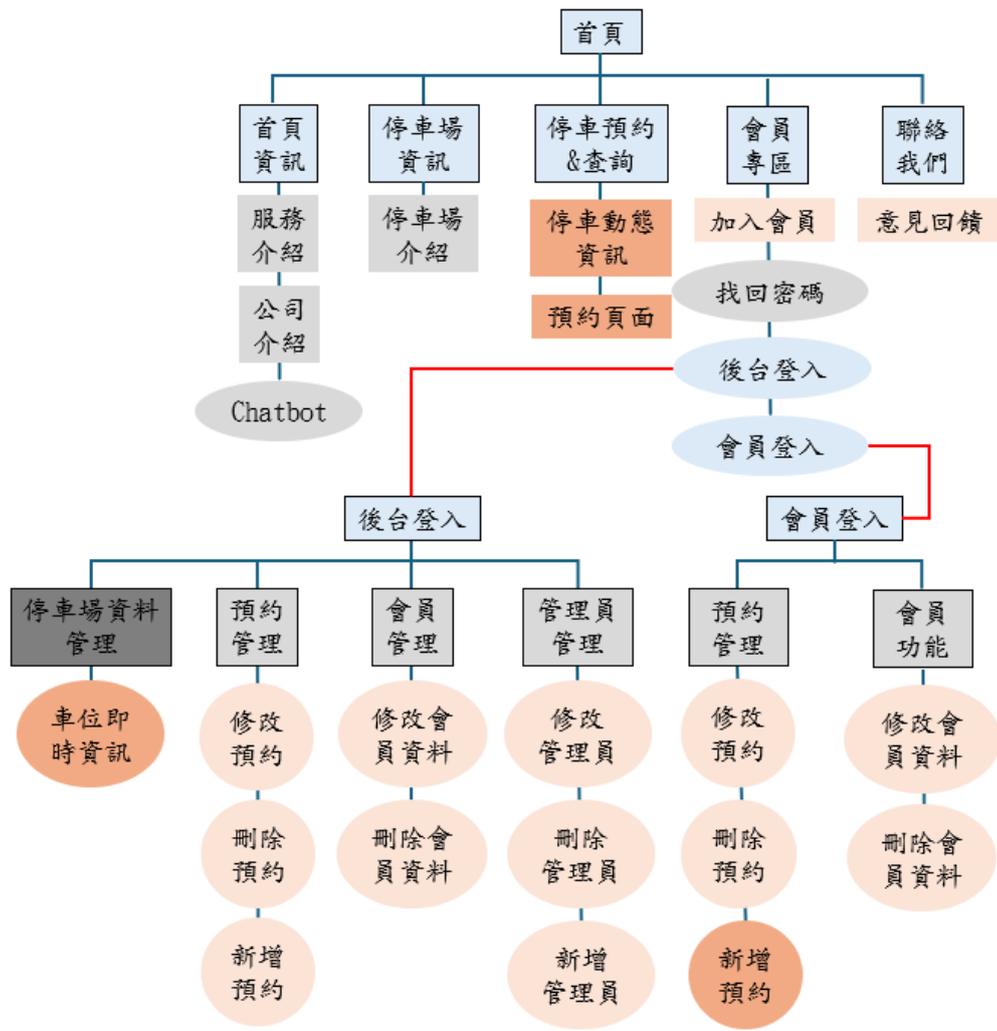


圖 12、網頁架構圖

4.2 ER-Model

本組利用 PhpMySQL 作為後端資料庫，並建立四個資料表儲存對應之資料，分別為會員資料表(member)、訂單資料表(order)、管理員資料表(manager)和停車場資訊表(parking lot)，其 ER-model 如圖 13 所示。

上述資料庫於期中專案報告後進一步的修正及改善，會員資料表中將會員細分為教職、學生及一般旅客等類別；訂單資料表則將車牌的資料納入於表中；停車表資訊表則是增加已預約量及預測預約量，可透過預約預測量作為會

員可預約車位數量之基準

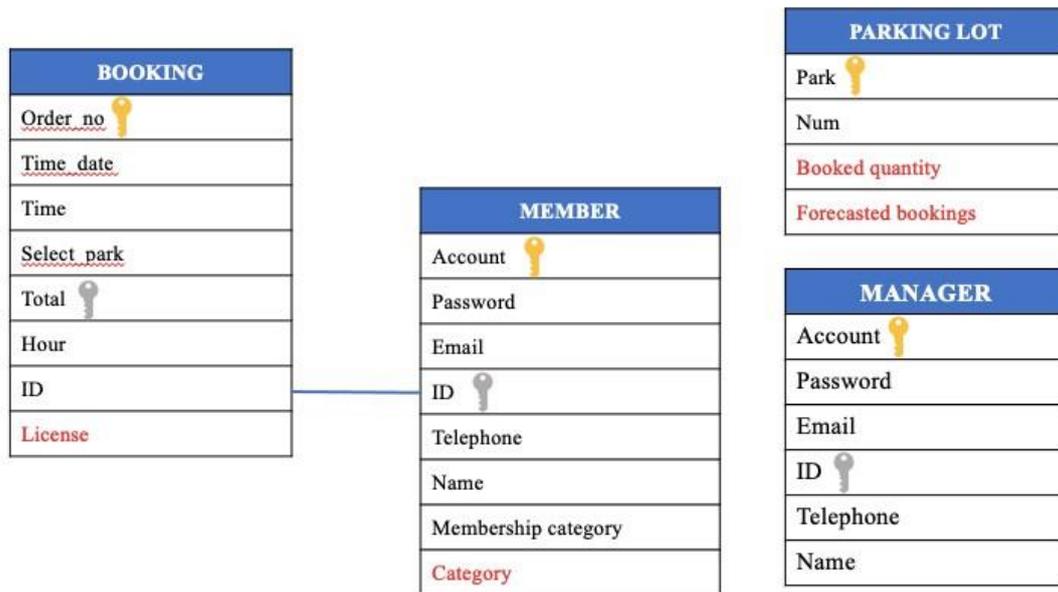


圖 13、ER-Model

4.3 功能介紹

4.3.1 網路預約及查詢

停車場用戶透過即時停車資訊的提供來進行停車位之預約，並透過填寫個人資訊及預約時間來完成預定工作，如圖 14 及圖 15 所示。



剩餘車位數即時資訊

若停車場車位容車量未滿，則可進行停車位之預約活動

機車		汽車	
行政大樓周邊停車場	剩餘預約量 20	<button>預約</button>	
工程一館旁邊停車場	剩餘預約量 10	<button>預約</button>	
風雲樓後方停車場	剩餘預約量 0	暫無空位	
大禮堂前面停車場	剩餘預約量 10	<button>預約</button>	
排球場旁邊停車場	剩餘預約量 30	<button>預約</button>	

圖 14、即時停車場資訊介面

車位預約

請填寫基本資料

填寫個人資訊

身份 學生 教職員 一般旅客

姓名

電話

車牌

日期

時間

訂票確認

圖 15、車位預約介面

4.3.2 會員專區

網站中設有會員專區，在會員專區的介面中除了可進行以進行會員登錄、加入會員和找回密碼之外，還能提供升級版之車位預約作業，如圖 16、圖 17

及圖 18 所示。預約完成後，即可於電子郵件收到預約成功之信件。

填寫個人資訊

身份 學生 教職員 一般旅客

 Lab724

停車系統預約

 1小時

停車系統預約

選擇日期:

(+08:00) CST - Taipei

< 六月 >

週一	週二	週三	週四	週五	週六	週日
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

選擇時間:



8:00 am
8:15 am
8:30 am
8:45 am
9:00 am
9:15 am
9:30 am
9:45 am
10:00 am
10:15 am
10:30 am

圖 16、會員預約介面

TidyCal ✕

確認預約:

停車系統預約 與 Lab724

📅 星期六, 六月 22日, 在 9:45 上午

Asia/Taipei

 1小時

您的姓名: 您的Email:

車牌

停車場

Park1 : 行政大樓周邊

Park2 : 大禮堂前

Park3 : 風雲樓後方

Park4 : 工程一館旁邊

This site is protected by reCAPTCHA and the Google Privacy Policy and Terms of Service apply.

圖 17、會員預約資料填寫

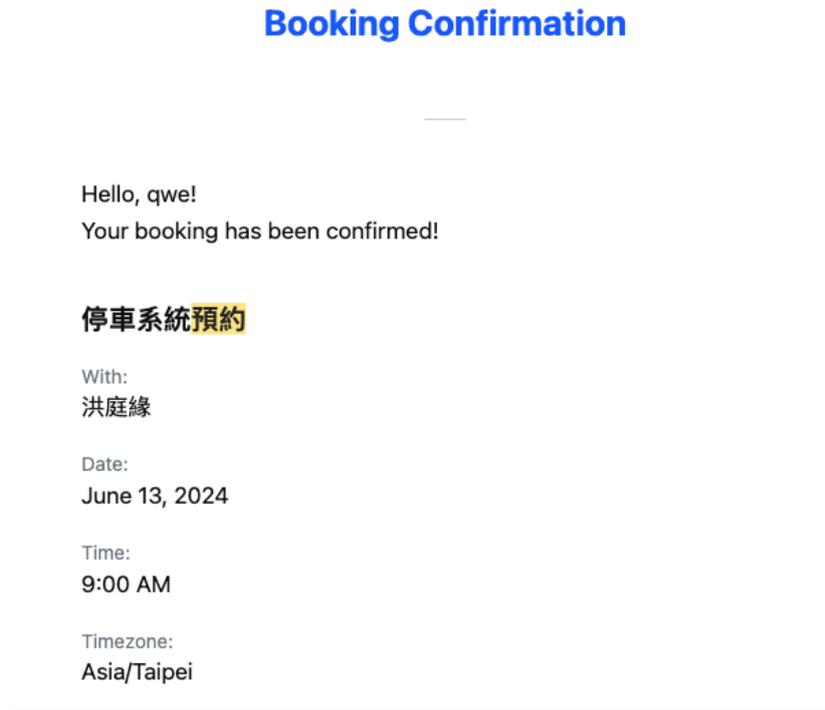


圖 18、預約成功確認信

4.3.3 後台管理

網站中設有管理員專區，在管理員專區的介面上登入管理員帳號後，即可進入後台管理系統，有預約資料管理、會員資料管理、管理員資料管理及停車場資料管理等四大功能區塊，皆可以查看現有的資訊，並直接對資料庫進行修改刪除(管理員資料管理額外增加新增管理員之功能)，其新增之功能可如圖 所示。其新增之功能可如圖 19 所示。

停車場資料

```
$row['park_name'], 'num' => $row['num'], 'reservation' => $row['reservation'], 'predict' => $row['predict'], ); } foreach($values as $v){ echo " "; echo " "; echo " ";  
echo " "; echo " "; echo " "; } //關閉資料連接 mysqli_close($link); ?>
```

停車場	總量	已預訂量	預估預約量
" . \$v['park_name'] . "	" . \$v['num'] . "	" . \$v['reservation'] . "	" . \$v['predict'] . "

圖 19、停車場資料列表

五、結論與未來展望

5.1 改善成果

本研究成功地利用 TSD-GRU 模型改善了清華大學校園停車位的預測和管理系統，經過比較 LSTM、BiLSTM 和 TSD-GRU 三種模型的預測結果，我們發現 TSD-GRU 模型在準確性上具有顯著優勢，其 MAPE、RMSE、RAE 和 R2 等指標均優於其他兩種模型，通過應用該模型，我們可以更精確地預測停車場的佔用率，使得用戶在停車時能夠更高效地找到空位，縮短尋找停車位的時間，從而減少校園內的交通擁堵，透過網站及應用程式，使用者可以即時查詢停車位資訊並進行預約，並通過線上繳費系統進一步提升了用戶的停車體驗和停車場的運營效率，此次改進不僅提升了停車管理的自動化水平，也優化了校園內的交通流暢度，為師生和訪客提供了更加便捷的停車服務。

5.2 未來展望

未來，我們計劃進一步完善和擴展停車位預約系統，以提供更全面的服務。引入即時停車位影像辨識系統，結合影像辨識技術實時更新停車位狀態，提供更加準確和動態的停車資訊。除了現有的線上繳費系統，我們將增加更多支付方式，如行動支付、電子錢包等，讓用戶在繳費時有更多選擇。透過這些改進措施，我們期望能夠進一步提升停車場管理的智慧化水平，為用戶提供更加便捷、高效的停車服務，最終實現校園交通管理的全面優化。

